CDF Run II Data File Catalog

J.Kowalkowski[1], S.Lammel[1], D.Litvintsev[1],
F.Ratnikov[2], T.Watts[2], E.Wicklund[1]

[1]Fermi National Accelerator Laboratory, Batavia, Illinois, USA
[2]Rutgers University, Piscataway, New Jersey, USA

Abstract

The CDF experiment started data taking in April 2001. The data are organized into datasets which contain events of similar physics properties and reconstruction version. The information about datasets is stored in the Data File Catalog, a relational database. This information is presented to the data processing framework as objects which are retrieved using compound keys. The objects and the keys are designed to be the algorithms' view of information stored in the database. Objects may use several DB tables. A database interface management layer exists for the purpose of managing the mapping of persistent data to transient objects that can be used by the framework. This layer exists between the algorithm code and the code which reads directly from database tables. At the user end, it places get/put interface on a top of a transient class for retrieval or storage of objects of this class using a key. Data File Catalog code makes use of this facility and contains all the code needed to manipulate CDF Data File Catalog from a C++ program or from the command prompt. It supports an Oracle interface using OTL, and a mSQL interface. This code and the Oracle implementation of Data File Catalog were subjected to tests during CDF Commissioning Run last fall and during first weeks of Run II in April. It performed exceptionally well.

Keywords: CDF, RDBMS, Databases

1 Introduction

CDF experiment analyzes events produced in proton-antiproton collisions at a center-of-mass energy of 2 TeV at Tevatron collider located at Fermilab. The Run II phase of experiment has started in the beginning of April, 2001. Data acquisition system defines about 50 primary datasets formed together into 8 streams. Data is recorded at the peak rate of 20 MB/s and is immediately available to CDF Data Handling (DH) system. The purpose of the DH system is to archive, organize and provide access to the data.

The Data File Catalog (DFC) is a crucial ingredient of the CDF Data Handling system. It consists of an Oracle relational database which keeps information about the datasets, and a C++ API – the DataFileDB package, which contains all the code necessary to manipulate the data stored in database from C++ program or from the command prompt.

The DataFileDB package is a part of the CDF software infrastructure. It relies on data base management software developed at CDF – the DBManager package. The latter provides database implementation independent API for mapping persistent data to transient C++ objects[1].

2 Data Hierarchy

Data collected at CDF are organized in hierarchical fashion. The lowest layer is runsection – set of events, collected during about 30 seconds of data taking. Runsection is the unit of luminosity bookkeeping. Data quality bits are also defined by runsection. Dynamic trigger prescales are recorded at runsection boundaries. Data files are collections of runsections. Files are aligned at runsection boundaries meaning that all events of a given runsections are contained in one file. A group of data files stored together in a tape partition is called a fileset. The number of files

in a fileset (currently about 10) is chosen to optimize tape I/O. The filset is the unit of data of disk management software. The next layer is a dataset and finally the data stream.

3 The Data File Catalog

Information, necessary to locate any CDF dataset is stored in Data File Catalog. It contains:

- four core tables, corresponding to four elements of data access hierarchy – `CDF2_DATASETS`, `CDF2_FILESETS`, `CDF2_FILES`, `CDSF2_RUNSECTIONS`
- general bookkeeping information like tape volume, contents and allocation – `CDF2_TAPES`
- physics related bookkeeping information such as data quality trigger and filter used, average and integrated luminosity, etc.

DataFileDB package has classes and utilities for inserting selecting and updating information in the database. There are five basic objects that define rows of various tables in the DFC: `DFCDataset`, `DFCFileset`, `DFCFile`, `DFCRunsection`, `DFCTape`. Each of these objects has interface which allows users to view information held inside the object. All five row objects have associated smart pointer objects. These are the objects returned from queries to database. Each of the row object has an accompanying key object. Several of row objects have hierarchical views associated with them. Hierarchical views make calls down to DBManager classes which perform connection to the database, do put, get and update queries. This is example code of getting information about the files which belong to dataset identified by string `"aphysr"`':

```
// make connection to database identified by key "production"
        DFCFileCatalogNode fc("production");
// key class associated with file
        DFCFileKey key;
        key.setDatasetNameID("aphysr");
// typedef std::vector<DFCFile> DFCFiles;
// typedef Handle<DFCFiles> DFCFiles_var;
        DFCFiles_var files;
        fc.findFiles(key,files);
```

`findFiles` method of hierarchical view `DFCFileCatalogNode` makes DBManager calls:

```
// typedef  Manager< DFCFiles,DFCFileKey > DFCFiles_mgr;
        DFCFiles_mgr m("production","DFCFiles");
        m.get(key,files);
```

DataFileDB package provides interface to DFC for DHInput and DHOuput modules – user friendly interfaces to CDF DH systems[2]

4 Conclusion

DataFileDB code and Oracle implementation of Data File Catalog were subjected to test during CDF Commissioning Run last fall and since the start of Run II. The system has been performing exceptionally well.

References

[1] J. Kowalkowski *et al.*, "Mapping Auxiliary Persistent Data TO C++ Objects in the CDF Reconstruction Framework" Published in CHEP 2000, Computing in high energy and nuclear physics 454-457

[2] F.Ratnikov *et al.*, "User Friendly Interface to the CDF Data Handling System" this conference